|  |
| --- |
| Project Title *Basic Word Processor* |
| Author: *Stephen Pollard* Testers: Daryl Winslow  Adam Noonan  Marcel Schilsky  Stephen Pollard |
| Version Number:4 **Change History**   |  |  |  | | --- | --- | --- | | **Date** | **Author** | **Comments** | | 23/3/15 | Stephen Pollard | The first change was due to the failing of the add image testing which was due to out-dated code, we were not able to implement the feature. | | 27/3/15 | Stephen Pollard | Spell checker caused a lot of trouble and hassle as It was a large chunk of code. When testing came around it proved our knowledge of python really became obvious and it just wasn’t comprehensive enough to get the feature to function properly. This prompted us to rearrange the code, omitting some classes and just making the code easier to read and understand. | | 28/3/15 | Stephen Pollard | Font was one of the main features we tried to implement. It posed many hurdles to get running properly that at the time we weren’t capable of overcoming. This changed some of the structure of the project. | | 21/4/15 | Stephen Pollard | Eventually, broadening our knowledge of python over the last few weeks we were able to give the font another go and finally got it functioning to near what we sought after in the beginning. This added a huge aspect to the overall project and added some huge important features that are critical to a word processor. | |
| Release Criteria *[Set with agreement with the team leader and checking back to the customer if necessary]*  A:Basic word processor able to open, edit and save txt documents.  B:Graphical user interface that is clear, simple and intuitive.  C:Additional features E.g. Word Count, Font formatting.  D:Ready to release by April 22nd. |
| Related Documents  |  |  |  | | --- | --- | --- | | Document Title | Author(s) | Description | | *Project plan* | Adam Noonan | Document outlines our plan to implement the desired outcome showing the different steps of approach to do with design, testing, implementation and documentation of the project. | | *Design document* | Team | Document specifies our intended approach and goal in terms of the design of the project. | | *Requirements document* | Team | Document showing our analysis of the requirements needed for the desired outcome. | | *Other documents* |  |  | |  |  |  | |  |  |  | |
| Test Strategy: *Our approach to testing was very simple. We used white box testing which is effectively a method of testing that utilises our programming skills to examine the output. As we knew the intended goal of the software we understand its functionality and how it was meant to work. This allowed us to play around with the code to find any errors and eventually attain the desired output. When we decided to use a programming language that we had minimal knowledge of, white box testing was always going to the method of testing because while learning the language during the process, this made testing very trial and error. We considered black box testing but as we had a time limit during the process, on closer inspection we came to the conclusion it would be too time consuming as we would have to recruit people outside of the project who didn’t know the ultimate goal or functionality of the software.* |
| Test Plan: *[Each test will be completed must be listed here. Give each test a number]*   |  |  |  | | --- | --- | --- | | Number | Module | Description | | 1  2  3  4  5  6  7 | Open File  Close  File  Find and replace  Add Image  Font/  Font colour  Word Count  Spell Checker | User must be able to open a document file for editing.  User must be able to close any file that is currently open or in use.  User can search for a certain word or sentence and replace it with desired words. These two are to be tested individually and together.  User can add an image or picture into their text document.    User must be able to change the font and colour of their text to what they desire.  The word processor must keep track of the word count typed by the user and return it to the user on the screen. Achieved by the use of a status bar.  The word processor should have a spell checker that corrects any misspelled words or grammar mistakes. | |
| Test Results Summary: *[Put the results of the final test prior to release]*   |  |  |  | | --- | --- | --- | | Type | Count | List of test numbers failed | | A | 3 | 4,5,7 | | B | 0 |  | | C | 1 | 6 | | D | 3 | 1,2,3 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 1 | Stephen | Marcel, Daryl, Adam, Stephen | 1/3/15 | Passed |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 2 | Stephen | Marcel, Daryl, Adam, Stephen | 5/3/15 | Passed |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 3 | Stephen | Marcel, Daryl, Adam, Stephen | 12/3/15 | Passed |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 4 | Stephen | Marcel, Adam | 23/3/15 | Failed | When trying to implement the add image functionality we  quickly discovered it was near impossible to get it working.  This was because we had chosen to use the latest version of  Python, 3.0. Once we ran our test it showed that python  3.0 did not support image libraries. Whereas 2.0 would have  Allowed us to easily implement this function. As we had a  huge bulk of our project already completed in 3.0, it simply  wasn’t feasible to redo the majority of our code to suit one  feature. | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 5 | Stephen | Stephen ,Marcel | 28/3/15 | Failed | At first when we implemented font, it was functioning  some what but not to where we wanted it to be. The reason  the test failed is because we had to hardcode the font into  the text widget. This didn’t allow the user to select the font  or font colour that they desired. | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 5 (Second attempt) | Stephen | Stephen ,Marcel,  Daryl, Adam | 21/4/15 | Passed | Note: Second attempt was successful and the project gained  Font and font size functionality. | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 6 | Stephen | Marcel, Daryl, Adam, Stephen | 23/3/15 | Passed |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Results:*[Each run of the tests should generate a new page]*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Number | Test Implemented by | Test Completed by | Date of completion | Success/failure | Description (if failure) | | 7 | Stephen | Marcel, Daryl | 27/3/15 | Failed | Spell checker was a very complicated feature to implement.  We tried using hash tables and dictionaries, even a basic  word list in the format of a text document. None of these  came to fruition. We even tried a roundabout way to  implement the spell checker through a simple save of the  text document just created and then opening that through  spell checker. | |